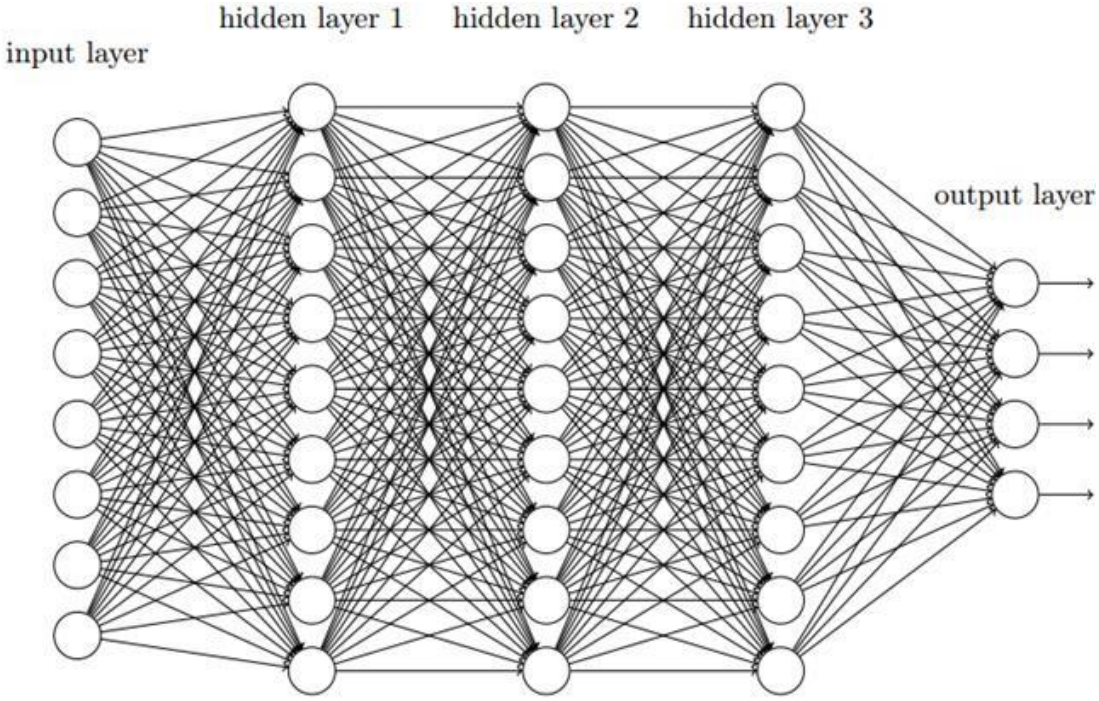
卷积神经网络（CNN）

全连接网络

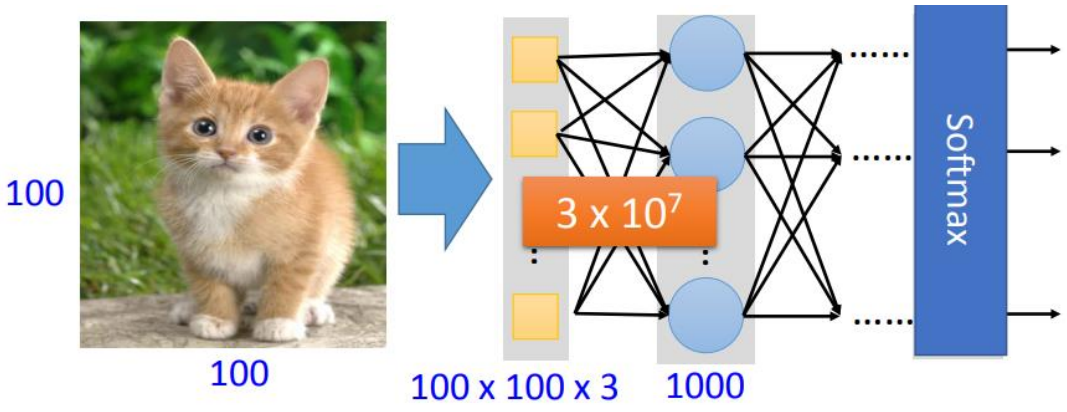
全连接网络是最简单的神经网络，即相邻层上的每个神经元都连接。



每条边都代表着权重，这些权重通过前向传播和反向传播进行学习。

卷积神经网络

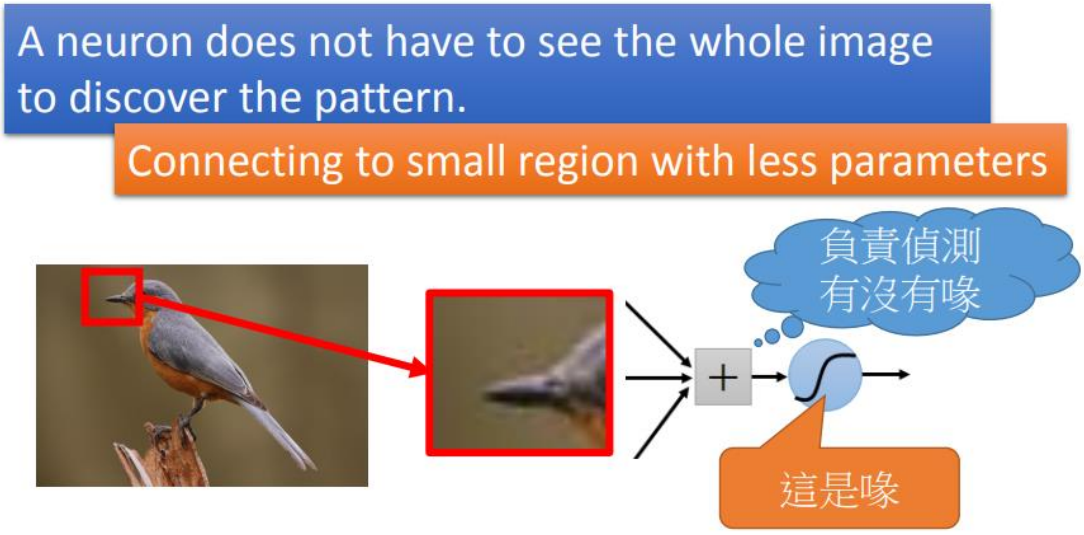
假设有一个目标识别的任务



我们使用全连接网络来学习，网络中的边也就是这个模型需要学习的参数非常多，导致学习速度会非常慢，这就需要引入一个更好的神经网络模型来提高学习速度。

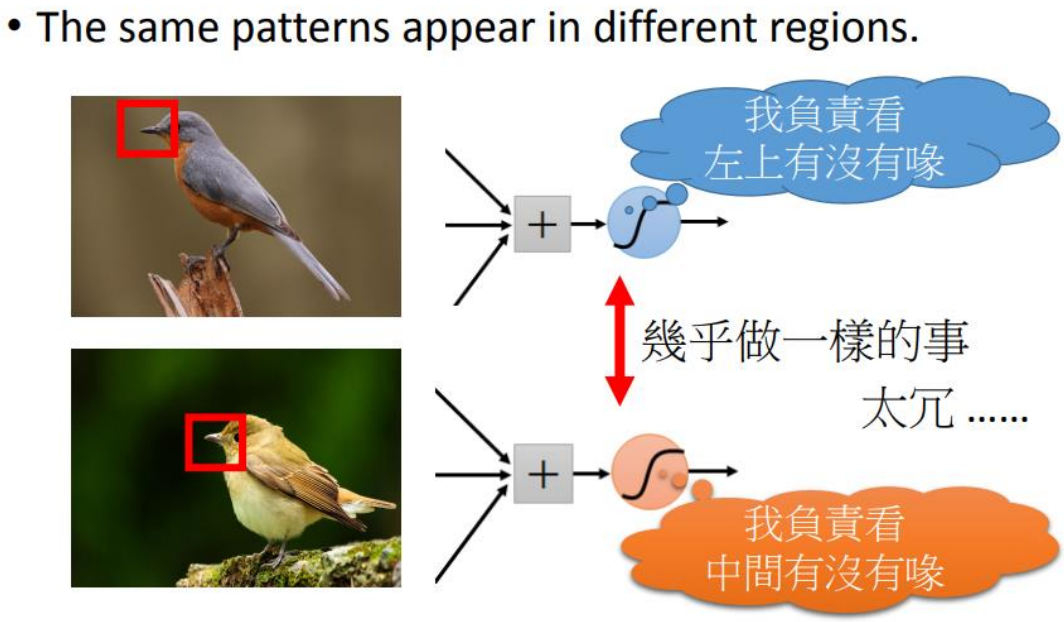
从图像的一些性质下手：

1. 图像上的特征通常都具有局部性，相对全局来说比较小，比如：

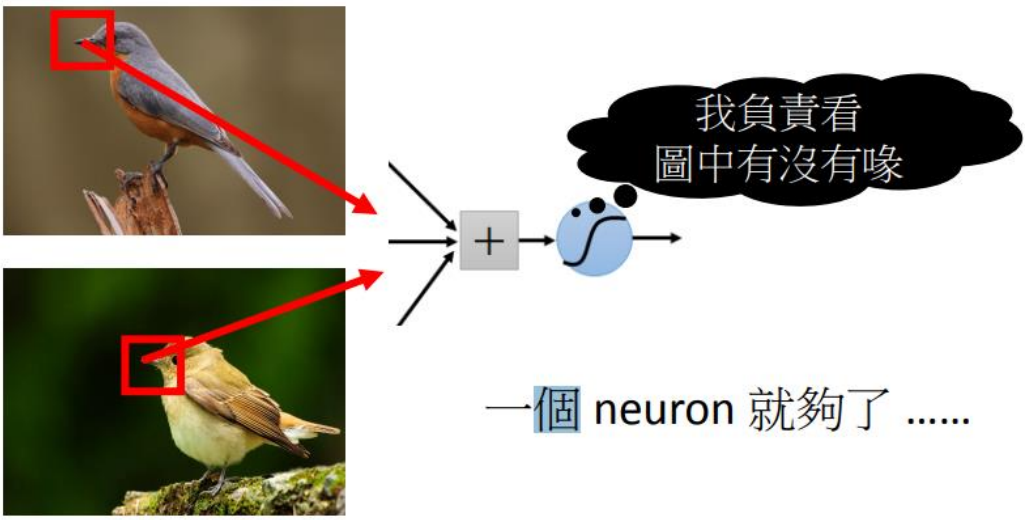


这样的话下一层的一个神经元没必要和前一层的所有神经元都连接。

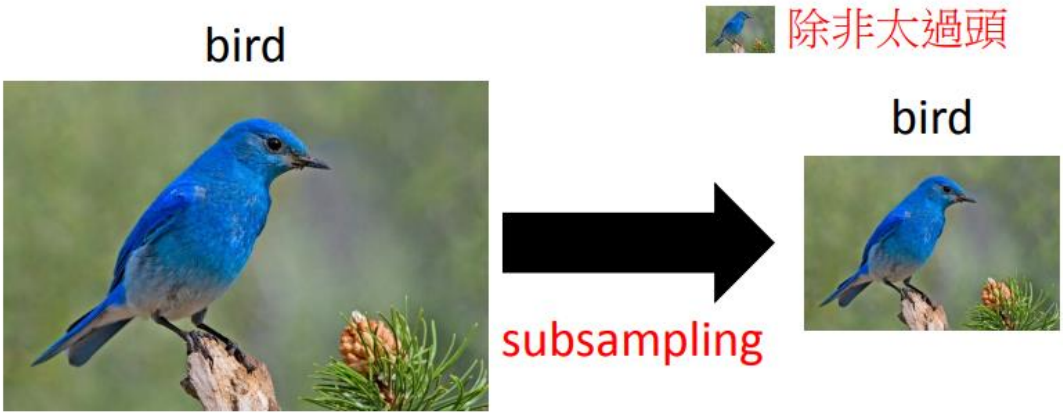
2. 同一个特征可能出现在图像上的不同空间位置，如果采用全连接网络，就导致也许会有某些节点他们的权重相近，这样就造成了冗余。



所以可以“共享权重”，如下：



3. 适当降采样图像不会改变目标的特征



降采样的好处在于既保持了特征没有发生较大变化，同时让下一层的输入变少了，那么学习的参数也就变少了。

根据上述 3 个图像的属性，提出了卷积层和池化层。

1. 卷积层

   主要有多个卷积核组成，每个卷积核对图像进行卷积操作，每个卷积核都不大，对图像扫一遍，对每一个子区域进行加权求和，这就满足了刚说的图像特征的**局部性**以及**共享权值**（扫一遍图像的过程其实就是图像的各个子区域共享了这个卷积核的参数）。

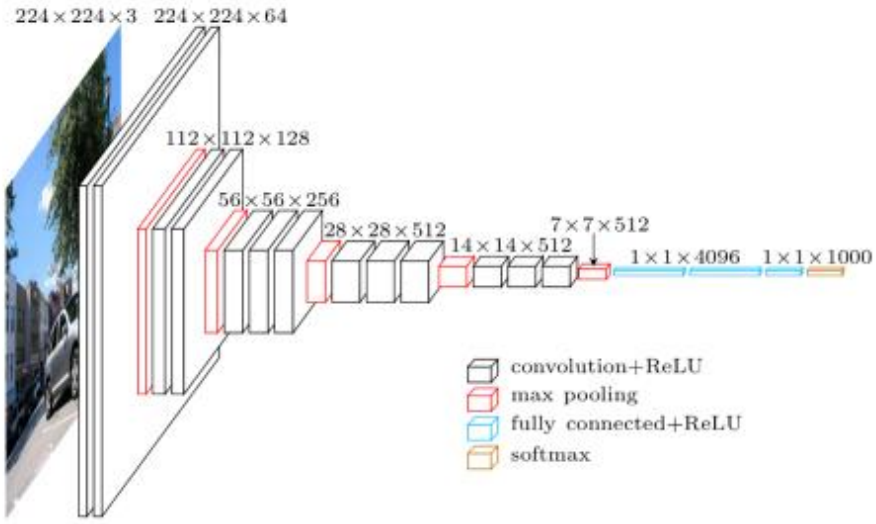
2. 池化层

   池化层用来实现**降采样**，常用的是最大池化层，类似卷积核，在图像上扫一遍，只不过简单一点，这个核不做卷积操作，只是把这个小区域内的最大值取出来，这样的话就能够把整张图片变小，比如原图像如果是 16x16 的图像，经过 2x2 的最大池化层后就变成了 8x8 的图像。

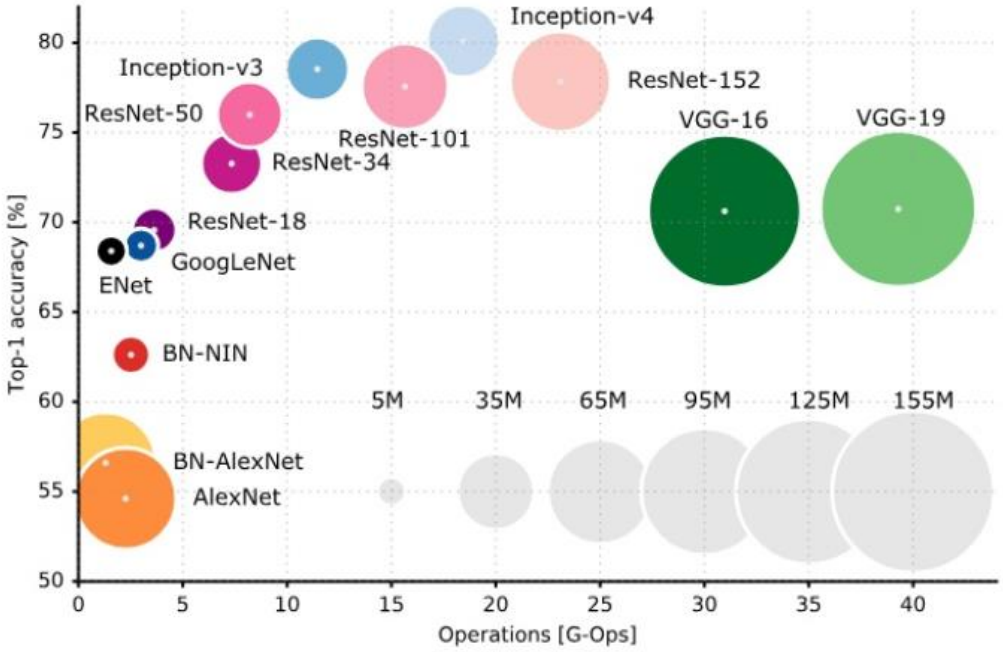
一个卷积神经网络通常就是先过一层卷积层，得到多个 Feature map；然后在通过池化层，执行降采样，让 Feature map 变小；当然前面都是考虑了局部特征，我们最后需要将多个 Feature map 整合在一起，需要 resize 成一个向量然后输入给全连接层，得

到最终的输出。总而言之，一个 CNN 通常由卷积层，池化层和全连接层组成。

不同的 CNN 区别在于它们的结构不同，比如 VGG-16:

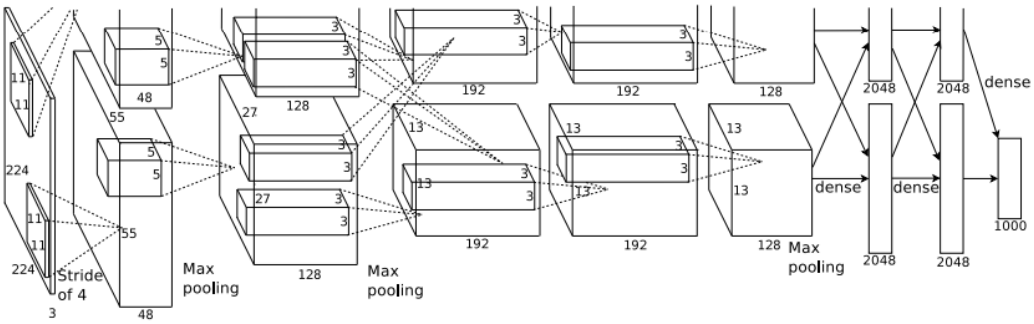


当然也有很多流行的 CNN:



论文里 CNN 的使用

论文的算法框架里，采用了《ImageNet Classification with Deep Convolutional Neural Networks》中的 CNN，拥有 5 个卷积层和 2 个全连接层:



本论文进行了微调，输入不再是上图中的 224x224 的 RGB 图像，改为 227x227 的 RGB 图像。论文中把这个网络称为 T-Net。

CNN 的结构对 R-CNN 的性能有所影响，论文后面的实验中尝试使用 VGG-16 的结构进行学习，进行微调后，论文里称为 O-Net，效果有所提高，见下表：

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN T-Net	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN T-Net BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
R-CNN O-Net	71.6	73.5	58.1	42.2	39.4	70.7	76.0	74.5	38.7	71.0	56.9	74.5	67.9	69.6	59.3	**35.7**	62.1	64.0	66.5	**71.2**	62.2
R-CNN O-Net BB	**73.4**	**77.0**	**63.4**	**45.4**	**44.6**	**75.1**	**78.1**	**79.8**	**40.5**	**73.7**	**62.2**	**79.4**	**78.1**	**73.1**	**64.2**	35.6	**66.8**	**67.2**	**70.4**	71.1	**66.0**

Table 3: Detection average precision (%) on VOC 2007 test for two different CNN architectures. The first two rows are results from Table 2 using Krizhevsky et al.'s architecture (T-Net). Rows three and four use the recently proposed 16-layer architecture from Simonyan and Zisserman (O-Net) [43].

采用 O-Net 学习，在 VOC2007 上进行测试的 mAP 提高到 66.0。

参考资料

李宏毅 Deep Learning Tutorial